

AD-A120 560 NETWORK TREE TECHNOLOGY: AS APPLIED TO DEVELOPMENT OF 171

AD-A120 560 NETWORK TREE TECHNOLOGY: AS APPLIED TO DEVELOPMENT OF 171

AD-A120 560 NETWORK TREE TECHNOLOGY: AS APPLIED TO DEVELOPMENT OF 171

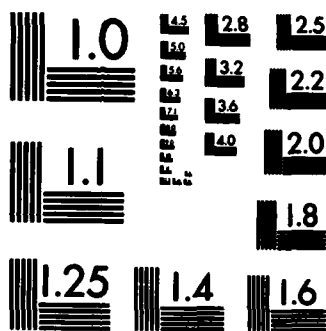
UNCLASSIFIED F33615-81-C-0010 F/G 9/5 NL

UNCLASSIFIED F33615-81-C-0010 F/G 9/5 NL

UNCLASSIFIED F33615-81-C-0010 F/G 9/5 NL

UNCLASSIFIED F33615-81-C-0010 F/G 9/5 NL

6310



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A120560

AFHRL-TP-82-29

(12)

AIR FORCE



HUMAN

RESOURCES

**NETWORK TREE TECHNOLOGY:
AS APPLIED TO DEVELOPMENT OF
FAULT ISOLATION PROCEDURES**

By

David E. Peterson

**The Boeing Company - Houston Operations
P.O. Box 58747
Houston, Texas 77058**

**LOGISTICS AND TECHNICAL TRAINING DIVISION
Logistics Research Branch
Wright-Patterson Air Force Base, Ohio 45433**

September 1982

DTIC
SEP 1982
A

Approved for public release; distribution unlimited.

LABORATORY

DTIC FILE COPY

**AIR FORCE SYSTEMS COMMAND
BROOKS AIR FORCE BASE, TEXAS 78235**

82 10 21 009

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Public Affairs Office has reviewed this paper, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This paper has been reviewed and is approved for publication.

DONALD L. THOMAS
Contract Monitor

JOSEPH A. BIRT, Lt Col, USAF
Technical Director, Logistics and Technical Training Division

DONALD C. TETMEYER, Colonel, USAF
Chief, Logistics and Technical Training Division

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|--|--|--|
| 1. REPORT NUMBER AFHRL-TP-82-29 | 2. GOVT ACCESSION NO. AD-A120560 | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle) NETWORK TREE TECHNOLOGY: AS APPLIED TO DEVELOPMENT OF FAULT ISOLATION PROCEDURES | | 5. TYPE OF REPORT & PERIOD COVERED Final |
| | | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s) David E. Peterson | | 8. CONTRACT OR GRANT NUMBER(s) F33615-81-C-0010 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS The Boeing Company - Houston Operations P.O. Box 58747 Houston, Texas 77058 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61101F ILIR0038 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS HQ Air Force Human Resources Laboratory (AFSC) Brooks Air Force Base, Texas 78235 | | 12. REPORT DATE September 1982 |
| | | 13. NUMBER OF PAGES 36 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Logistics and Technical Training Division Logistics Research Branch Air Force Human Resources Laboratory Wright-Patterson Air Force Base, Ohio 45433 | | 15. SECURITY CLASS. (of this report) Unclassified |
| | | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |
| 16. DISTRIBUTION STATEMENT (of this Report) Approved for public release, distribution unlimited. | | |
| 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) | | |
| 18. SUPPLEMENTARY NOTES | | |
| 19. KEY WORDS (Continue on reverse side if necessary and identify by block number) | | |
| computer technology data record forest functional pathing ground forest | key element logic diagram maintenance network tree nodal set | power forest reference designator signal forest sneak circuit analysis |
| 20. ABSTRACT (Continue on reverse side if necessary and identify by block number) | | |
| <p>→ This paper describes the activities and conclusions of the Air Force study contract designed to investigate the application of network tree technology to the development of Fault Isolation Procedures. Network tree technology, which was initiated as a formal electrical analysis technique known as Sneak Circuit Analysis, requires that the exact electrical configuration of a system be resident in a computerized data base. This study was to determine how this data base can be used to aid the fault isolation process. Through the analysis of a specific test case, the details of a formal Fault Isolation Procedure development methodology were developed along with the definition of key areas in which automation could be used to make this methodology more cost effective.</p> | | |

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Item 20 (Continued):

The results of this study show that the task of developing Fault Isolation Procedures is greatly aided by the use of a computerized data base that depicts the exact electrical continuity of the target system. This data base is selectively sorted to produce plots of system continuity at various levels of detail, and each plot is then used as a tool to create the fault isolation logic.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

**NETWORK TREE TECHNOLOGY:
AS APPLIED TO DEVELOPMENT OF
FAULT ISOLATION PROCEDURES**

By

David E. Peterson

**The Boeing Company - Houston Operations
P.O. Box 58747
Houston, Texas 77058**

**LOGISTICS AND TECHNICAL TRAINING DIVISION
Logistics Research Branch
Wright-Patterson Air Force Base, Ohio 45433**

Reviewed by

**Robert C. Johnson
Chief, Maintenance Performance Section
Logistics Research Branch**

Submitted for publication by

**William B. Askren
Senior Scientist
Logistics Research Branch**

PREFACE

This technical paper was prepared by the Boeing Company, Houston Operations, Houston, Texas. It describes work accomplished under contract F33615-81-C-0010 for the Air Force Human Resources Laboratory, Logistics and Technical Training Division. Mr. David E. Peterson was the principal investigator for the effort. Mr. Edwin G. McFall directed the effort for the Laboratory. Funding for the effort was provided from Laboratory Director's funds.

Accession For
NYTS GRAF
ERIC TAN
K...
J...

DTIC
COPY
INSPECTED
2

TABLE OF CONTENTS

| Section | Title | Page |
|---------|--|------|
| 1 | SCOPE | 5 |
| | 1.1 Background | 5 |
| | 1.2 Goals and Objectives | 5 |
| | 1.3 Ground Rules and Assumptions | 6 |
| 2 | STUDY EFFORT SUMMARY | 7 |
| | 2.1 Network Tree Sorting | 7 |
| | 2.2 Forest Development | 8 |
| | 2.3 Test Case Analysis | 8 |
| 3 | FAULT ISOLATION PROCEDURE DEVELOPMENT | 14 |
| | 3.1 Top Level Logic | 14 |
| | 3.2 Second Level Logic | 15 |
| | 3.3 FIP Development (LRU Level) | 16 |
| | 3.4 FIP Development (Internal to an LRU) | 16 |
| 4 | FIP DEVELOPMENT METHODOLOGY | 22 |
| | 4.1 Data Base Development | 22 |
| | 4.1.1 Data Requirements | 22 |
| | 4.1.2 Data Entry Process | 22 |
| | 4.1.3 Change Tracking | 23 |
| | 4.2 Forest and Network Tree Development | 23 |
| | 4.2.1 Data Base Sorting | 23 |
| | 4.2.2 Forest and Tree Plotting | 23 |
| | 4.3 Key Element Analysis | 24 |
| | 4.3.1 Fault Indicator Analysis | 24 |
| | 4.3.2 Input Element Analysis | 24 |
| | 4.3.3 Systems Operation Analysis | 24 |
| | 4.4 Logic Diagram Development | 24 |
| | 4.4.1 Forest Modification | 24 |
| | 4.4.2 Testing Logic Development | 25 |
| 5 | CONCLUSIONS | 26 |
| | 5.1 Study Results | 26 |
| | 5.2 Cautions | 26 |
| | 5.3 Technique Application | 27 |

TABLE OF CONTENTS

| Section | Title | Page |
|------------|------------------------------------|------|
| Appendix A | | |
| | DEFINITIONS OF KEY TERMS | 28 |
| | Data Record | 28 |
| | Nodal Set | 28 |
| | Network Tree | 28 |
| | Forest | 28 |
| | Power Forest | 29 |
| | Ground Forest | 29 |
| | Signal Forest | 31 |
| | Key Element | 31 |
| | Reference Designator | 31 |
| | Functional Pathing | 31 |
| | Logic Diagram | 31 |

LIST OF ILLUSTRATIONS

| Figure | Title | Page |
|--------|--|------|
| 1 | Reset Switch Signal Distribution | 9 |
| 2 | ELEC and SELECT Reset Signal Distribution Forests | 10 |
| 3 | ELEC and SELECT Reset Signal Common Point Application Example | 11 |
| 4 | LHT Servo Actuator/Select Reset Forest | 12 |
| 5 | Select Reset Relay Functional Forest | 13 |
| 6 | FLCS Fault Isolation Roadmap (Partial) | 14 |
| 7 | Pitch Axis Fault Isolation Roadmap (Partial) | 15 |
| 8 | Functional Forest for Pitch Lamp and Reset | 17 |
| 9 | Pitch Axis Fault Isolation Procedure 2740-A1, Page 1 | 18 |
| 10 | Pitch Axis Fault Isolation Procedure 2740-A1, Page 2 | 19 |
| 11 | Pitch Axis Fault Isolation Procedure 2740-A1, Page 3 | 20 |
| 12 | FLCC Internal Pitch Lamp Drive/Reset Forest | 21 |
| A-1 | Structure of an SCA Data Record | 28 |
| A-2 | Pitch Lamp Drive Circuitry Network Tree | 29 |
| A-3 | Pitch Lamp Drive Overview Forest | 30 |

NETWORK TREE TECHNOLOGY: AS APPLIED TO DEVELOPMENT OF FAULT ISOLATION PROCEDURES

1. SCOPE

1.1 Background. The rising cost of systems maintenance, the need for a high spares inventory, and a decrease in system availability, have necessitated research into techniques that could cost effectively produce Fault Isolation Procedures (FIPs) designed to combat these problems. This technical paper details the results of the study effort, performed for the Air Force Human Resources Laboratory (AFHRL), that examined the use of network tree technology, and its associated Sneak Circuit Analysis (SCA) data base, as a tool for the development of FIPs.

In 1967, Boeing initiated the development of the SCA technique. The goal of SCA was to discover basic design flaws in electrical systems without regard to system failure. This technique uncovers paths of undesired power/signal flow (sneak paths), timing problems (sneak timing), and false or misleading system indications and nomenclature (sneak indications and sneak labels). The SCA technique has been modified and updated continuously since its inception in order to keep pace with advancing electronic technology. In its present form, it is highly automated. Analysts take detailed system schematics, wire lists, and wiring diagrams and enter all of these data into a computer data base. Each data record explicitly describes the point-to-point continuity within the system being analyzed. After a series of audit and error check software has screened the data base, a series of reports are generated and translated into tree structured images (network trees) of the system's circuitry. Analysts then apply a series of clues (design checks) against these network trees, depending on the topology of the trees, in an effort to uncover sneak circuits. As the analysis of the trees progresses, reports are generated which describe sneak circuits, design concerns (questionable design practices), and document errors. New capabilities have been added to this technique, including the analysis of software. Compatibility of the network tree formats provides an effective tool for the analysis of hardware/software interaction, as well. The data base generated for a SCA also supports a variety of other analyses, such as Fault Tree Analysis and Failure Modes and Effects Analysis.

1.2 Goals and Objectives

1.2.1 Investigate the application of network tree technology for use in the development of FIPs. A goal of this study was to determine how this data base and the resultant network trees could be applied to the development of FIPs. Sections 2 and 3 describe how the data base and network trees were successfully applied.

1.2.2 Develop a formal methodology. A step-by-step approach should be developed for applying the network tree technology to FIP development; this methodology should include data base construction, manipulation, and network

tree construction and application. The formal methodology developed during this investigation is described in Section 4.

1.2.3 Identify areas for automation. The actual system configuration is resident in a computer, which can be a powerful tool for automating portions of the FIP development process and can improve the cost effectiveness of the FIP development task. The goal here was to identify those areas in which automation could be applied. The areas identified for automation are included in the formal methodology description (Section 4).

1.2.4 Provide a methodology to help alleviate current maintenance problems. A major goal was to produce FIPs which would help alleviate such maintenance problems as low system availability and high maintenance costs. In order to do so, these FIPs must minimize fault isolation steps required to isolate problems to the Line Replaceable Unit (LRU) while maximizing the use of system fault indication feedback. The FIPs must also provide the user with all of the pertinent information required to do the job, in a format which is readily understood. The example FIPs given in Section 3 illustrate the efficiency of the procedures developed using the network tree approach for both line and shop maintenance.

1.3 Ground Rules and Assumptions. This study effort was conducted using the following ground rules and assumptions.

1.3.1 F-16 Test Case. The study centered its system analysis phase on the F-16 aircraft. Two specific test cases were included in the task and were specified by the Headquarters, Tactical Air Command. The hardware involved included the pitch axis lamp and the servo/electronics reset switch and the circuitry directly related to these two components. The two cases identified were as follows:

1. Case 1: Problem - The F-16 is airborne when the pitch axis lamp illuminates. The lamp cannot be reset in flight.
2. Case 2: Problem - the F-16 is on the runway and the pitch axis lamp is illuminated. The lamp is successfully reset.

No other malfunction indications had been defined.

1.3.2 Use existing network trees. Network trees had previously been developed for the F-16 Flight Control System (FLCS) and the Electrical Power System (EPS). These network trees were to be used as the source of systems data for this study. It was recognized that:

1. The existing network trees might not totally reflect the present system configuration due to modifications made to the F-16 between the time the SCA was performed (1978) and the initiation of this effort (May 1981).
2. The network trees might not be in the optimum format for this task. In the performance of SCA, the location of

electrical components is not relevant to the analysis, and as such, network trees are constructed to illustrate the system circuitry without regard to physical location. Wiring data and connector and pin data are suppressed. It was envisioned that these data should not be suppressed for FIP development and also that the construction of the network trees should take physical location into account. All the encoded information would be available for analysis, however, even if not in the most optimum format.

3. All of the required data might not be available in network tree format. An SCA contract can be scoped to include only those subsystems as selected by the contracting agency. In the F-16 FLCS SCA task, certain modules, such as the Air Data Computer, were intentionally exempted from analysis. Should any of this information be required for this study effort, it would not be available in network tree format, and this fact would have to be noted.

1.3.3 Study to be performed manually. The study, using a "paper data base" would be performed manually, with no additional funding available for computer support. Areas for automation would be identified in support of future activities.

1.3.4 Fault Isolation Procedure development was not the primary objective of this study. The development of the methodology of applying network tree technology to FIP development was as important, if not more so, than the development of the actual procedures for the stated test cases. Any procedures developed under this contract, however, should reflect the benefit of the approach. These procedures did not have to comply with current military standards and specifications for FIPs.

2. STUDY EFFORT SUMMARY

This section of the technical paper summarizes the study effort activities. Its purpose is to define the flow and accomplishments of this study along with the points of caution recognized along the way. There are some major points of departure between this study effort and the actual methodology required to develop FIPs, based on network trees, and these points are also discussed here.

2.1 Network Tree Sorting

There were 1,980 existing F-16 network trees for the FLCS and the EPS (about 2,500 pages). Not all of these network trees were required by this study effort, and so this paper data base had to be sorted in order to extract those network trees which were applicable. This initial sorting activity, performed manually, resulted in the extraction of 613 network trees.

This sorting activity is the first point of departure from the FIP development methodology described in Section 4. The formal methodology requires that the system data be resident in the computer in the data record format previously

described. Since this study was performed using a paper data base (reference Section 1.3.2), the sorting activity required may not be necessary (at least to this extent) when using a computer-supported data base.

2.2 Forest Development

The next step in the study was to develop forests from the network trees extracted during the first study phase. Sixty-three forests were developed in the time available for this activity.

It became evident that forest development would be much more cost effective when performed using a computer supported data base. Manual forest development from existing network trees, therefore, represents another departure from the FIP development methodology given in Section 4. In essence, an active SCA data base should be used to dynamically create forests and network trees in the same time frame, rather than working back from existing network trees.

This forest development activity also pointed out the need for partitioning the data base in support of specific FIP development requirements. This is discussed in Section 4.

2.3 Test Case Analysis

The test case conditions (reference Section 1.3.1) required that the circuitry associated with the pitch axis lamp and the servo/electronics reset switch be analyzed. The reset function and the signals which cause the pitch lamp to illuminate had to be studied in order to assess how they interact. Reset signal distribution and application were analyzed. Analysis was required to determine if the reset signal could be interrupted due to system design, and if so, under what conditions this could occur.

The reset signals that were studied were generated when the servo/electronics reset switch on the Flight Control Panel (FLCP) was placed (momentarily) to the ELEC position. Figure 1 illustrates a portion of the network tree which shows this reset signal distribution.

The two signals of interest were the ELEC RESET signal and the SELECT RESET signal. Although both signals emanate from the same source, the SELECT RESET signal path contains a (normally closed) relay that would serve to interrupt this signal under certain conditions. Before investigating these conditions, however, the distribution of both these signals had to be traced in order to analyze how they are applied to the flight control circuitry. In order to do this, two forests were developed which showed the reset signal distributions. These forests are illustrated in Figure 2. The SELECT RESET signal, it turned out, was applied to the same network trees as the ELEC RESET signal with only five exceptions to which ELEC RESET was not applied. These exceptions were two network trees which dealt with the left and right flaperon logic: one which dealt with the rudder, and two which involved the left and right horizontal tail. Since the left and right horizontal tail circuitry can cause the pitch light to illuminate, these relations were studied further.

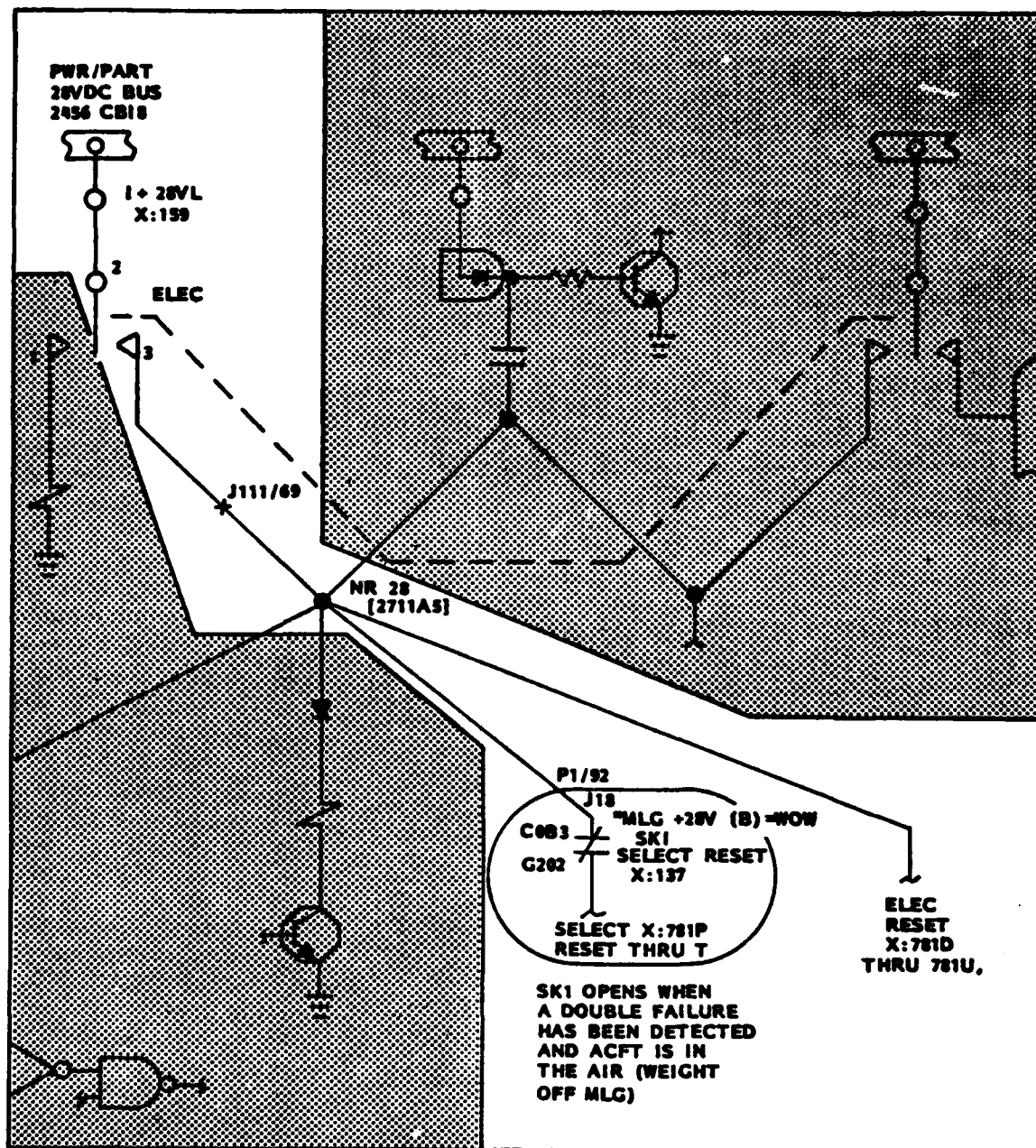


Figure 1. Reset Switch Signal Distribution

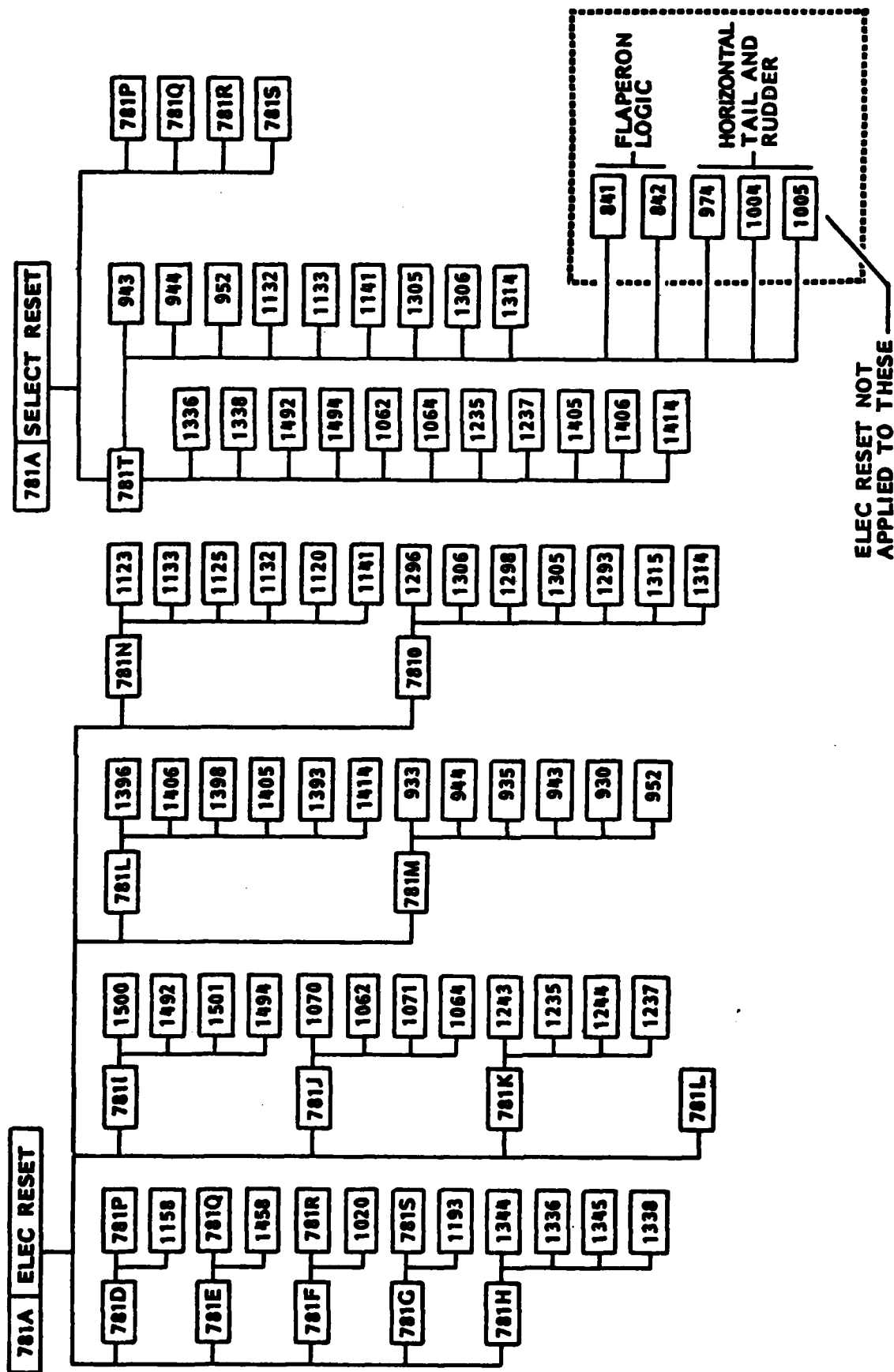


Figure 2. "ELEC" and "SELECT" Reset Signal Distribution Forest

It could not be assumed, in those cases where the SELECT RESET and ELEC RESET were applied to the same network tree, that they were applied in the same manner or even to the same circuitry. A quick study of the network trees involved, however, showed that these signals were applied to the same point (Figure 3). A loss of the SELECT RESET signal to these points would not result in the loss of the reset function.

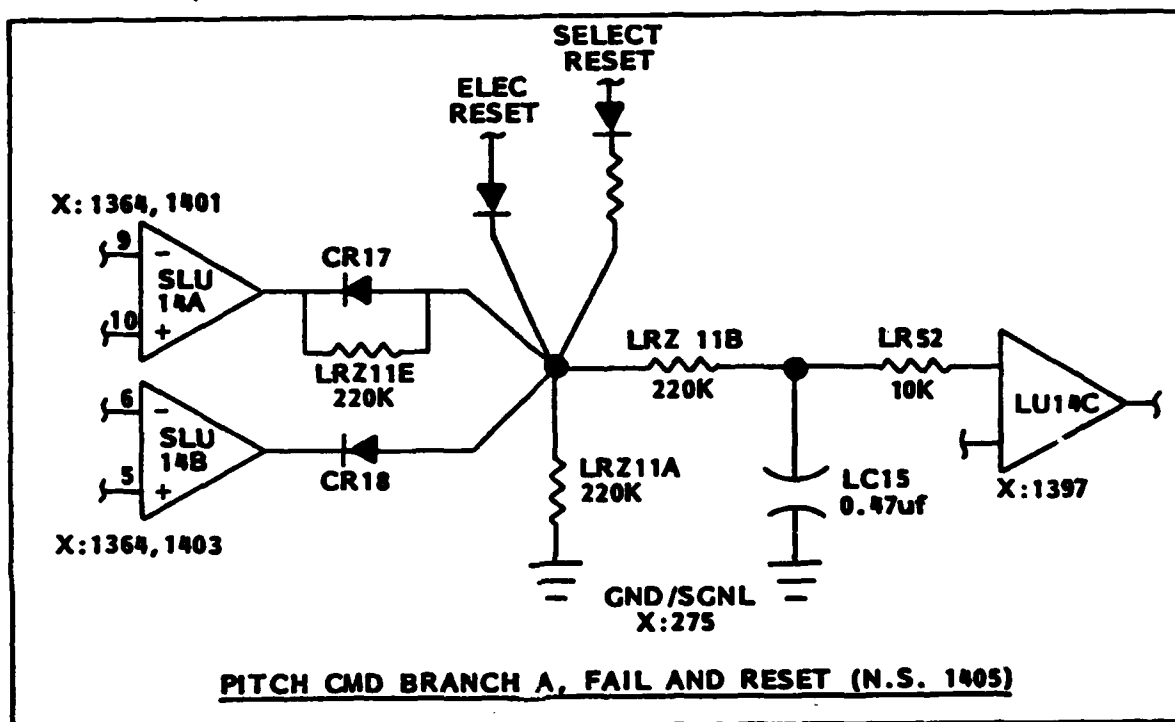


Figure 3. ELEC and SELECT RESET Signal Common Point Application Example.

Top level functional forests were developed for the pitch lamp drive circuitry and the select reset function. Figure 4 is the example for the left horizontal tail (LHT), and shows that a loss of the SELECT RESET signal could result in the inability to reset the pitch axis lamp. A similar forest exists for the right horizontal tail.

The network trees were then examined in order to determine the conditions under which the SELECT RESET signal would normally be interrupted.

As can be seen in Figure 5, two conditions must exist in order to interrupt the SELECT RESET signal (other than a signal path discontinuity). A dual failure condition (FC) must exist and the aircraft must be airborne. Other failure indications would also be present, although they were not defined in the original test case description. The Dual FC Fail warning lamp, for example, should also light under these conditions.

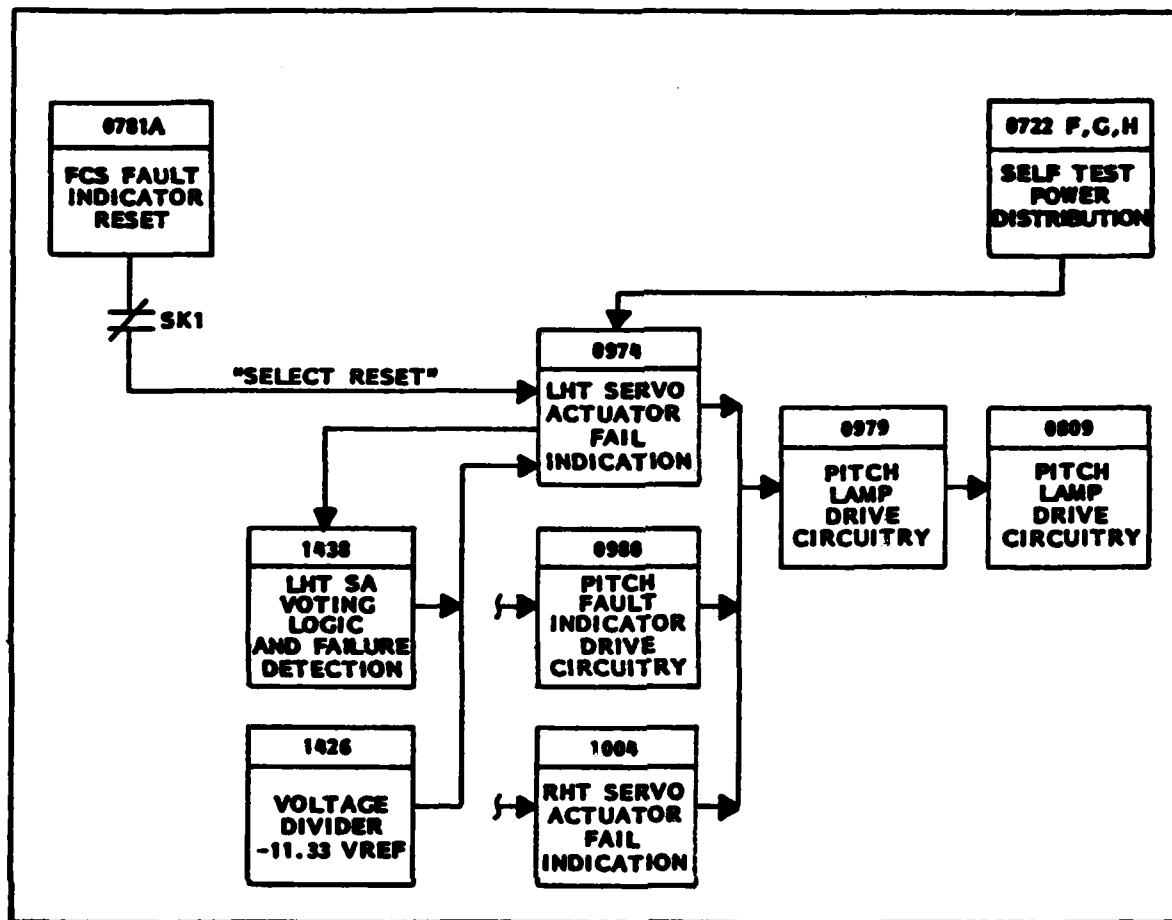


Figure 4. LHT Servo Actuator/Select Reset Forest.

Since a dual alpha fail (high) signal can also result in the opening of the select reset signal relay, the angle-of-attack (AOA) circuitry was also analyzed by the generation of AOA signal forests. During this analysis, it was noted that signals from Channel A of both AOA sensors were fed into the Air Data Computer. Since the Air Data Computer had not been included in the original SCA Statement of Work, there were no network trees available to continue this line of investigation.

This lack of data underscores a highly important point: all of the hardware in a system must be encoded and analyzed if effective FIPs are to be developed. Missing data could result in inadequate procedures because all of the possible cause and effect relationships cannot be fully defined. Vital fault indications could also go unrecognized, and it is a goal of network tree supported FIPs, to take full advantage of all available clues and indications in the fault isolation process.

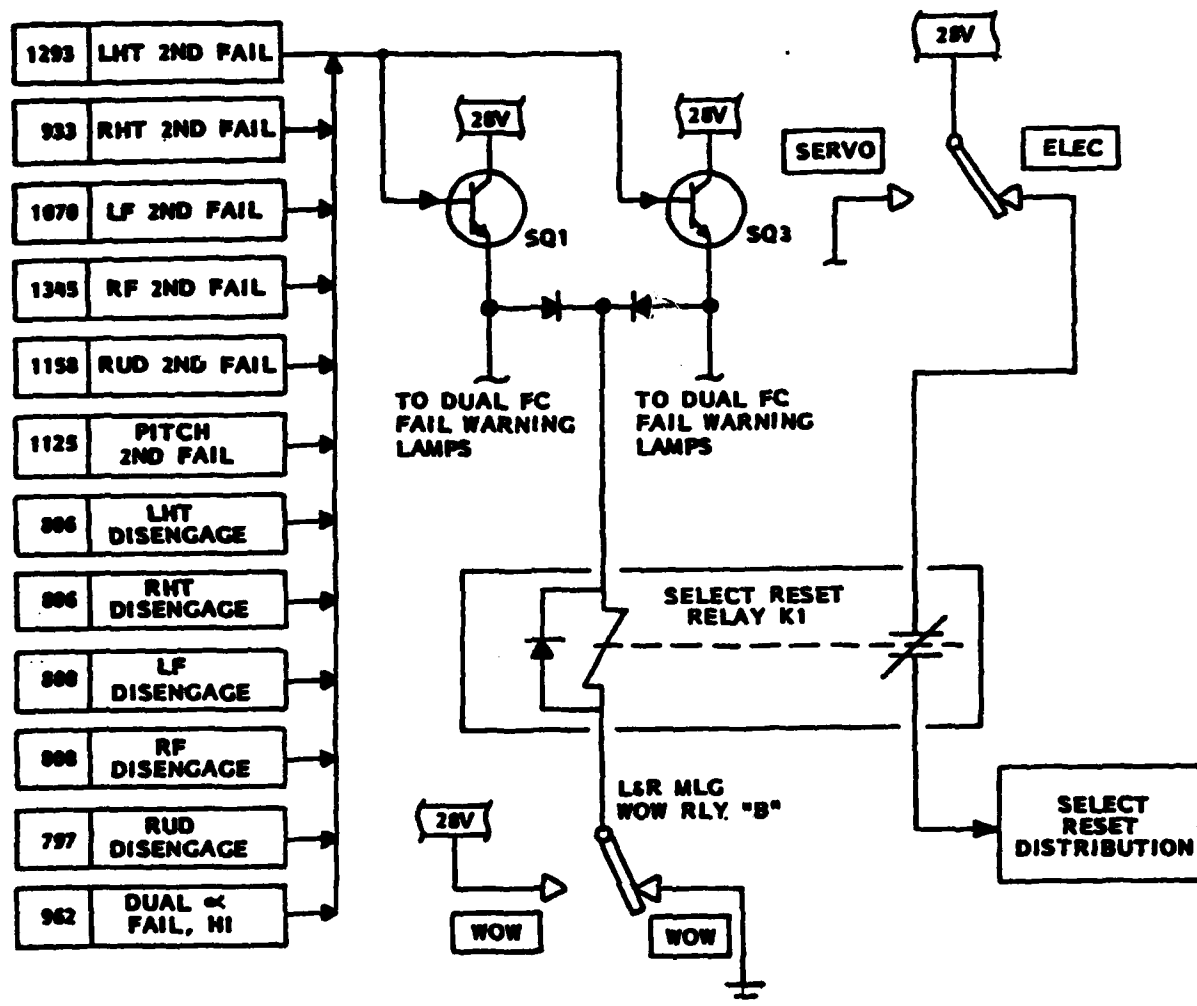


Figure 5. Select Reset Relay Functional Forest.

In the time available for test case analysis, the following conclusions were reached:

- a. The conditions existing in the test case can occur due to system design.
- b. The pitch axis lamp cannot be reset when
 - (1) the aircraft is airborne
 - (2) a dual failure condition exists involving the left horizontal tail, the right horizontal tail, or possibly a dual failure in the AOA circuitry.

- c. Case 2 (pitch light resets when the aircraft is on the ground) is a subset of Case 1 (pitch light cannot be reset while airborne) and also reflects system operation as designed.

3. FAULT ISOLATION PROCEDURE DEVELOPMENT

3.1 Top Level Logic. Effective FIPs will take advantage of all possible system clues in guiding the users to the appropriate test procedures. Multiple fault indications can also provide excellent pointers to specific system problems. At the top level, the goal is to present enough fault indication information to guide the user to the appropriate procedural group (in this case, pitch axis related failures). No testing should be required at this level. A partial top level logic diagram, the FLCS FAULT ISOLATION ROADMAP, is illustrated in Figure 6. It should be noted here that the "condition" column is not limited to physical fault indication clues. Sensory feedback, in this case to the pilot, can also be included in this column. In this example, the roadmap was not continued beyond the point where the next logic level (PITCH AXIS ROADMAP) was selected.

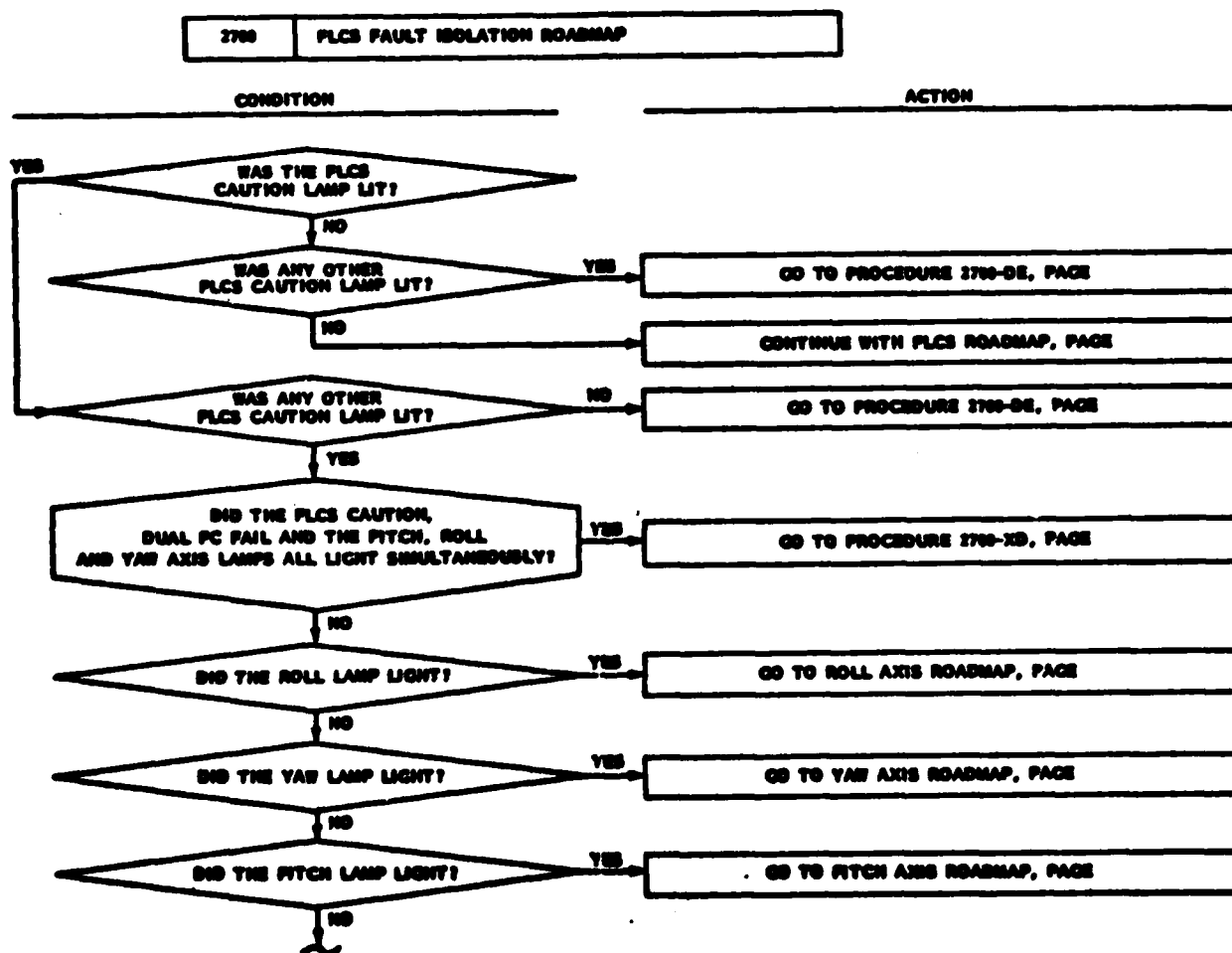


Figure 6. FLCS Fault Isolation Roadmap (Partial).

3.2 Second Level Logic. A second level of logic is required to guide the user from a procedural group to the specific FIP to be performed. Again, maximum use of the available clues must be made, and ideally no testing should be required at this level. The example illustrated in Figure 7 is a partial PITCH AXIS ROADMAP, and the logic involves only the data used in this test

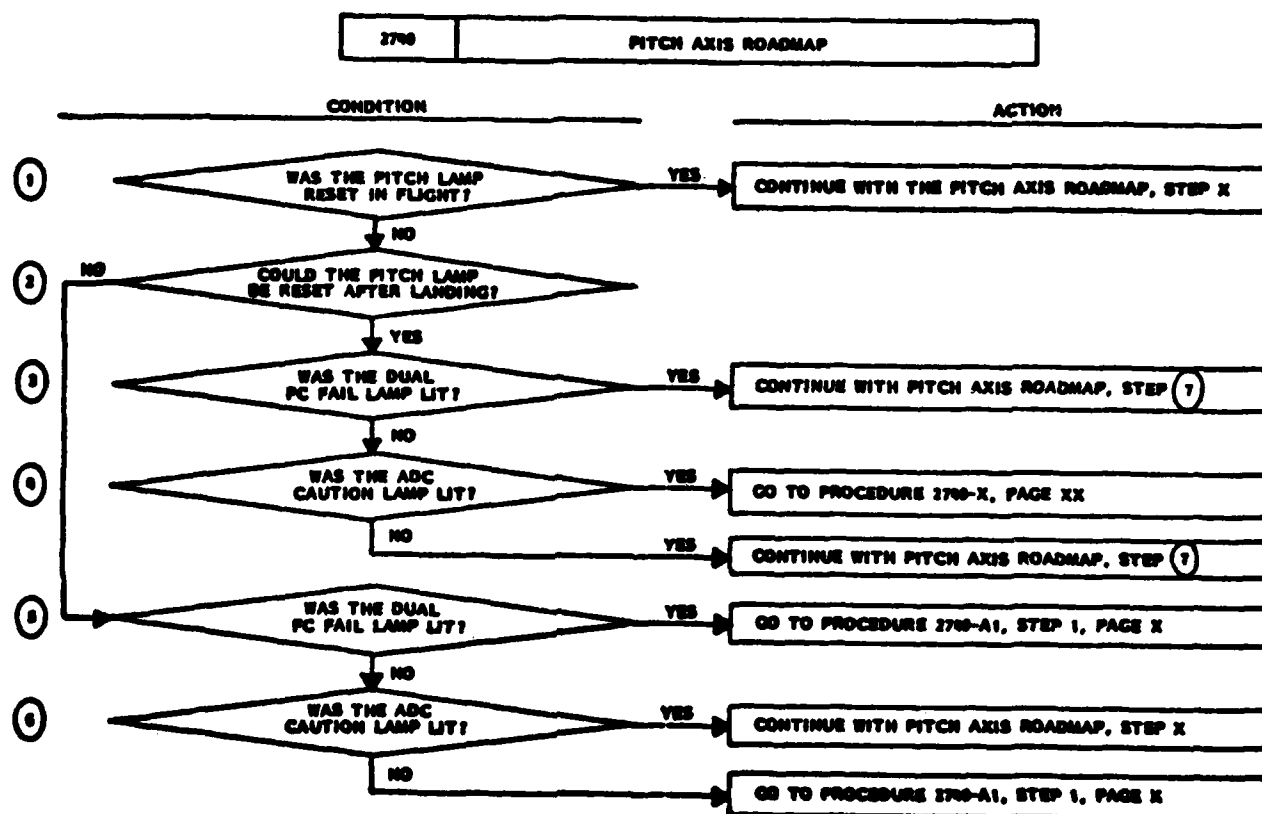


Figure 7. Pitch Axis Fault Isolation Roadmap (Partial).

case analysis. When structured to involve all of the possible conditions and fault indications which relate to the pitch axis, this logic could be quite different.

In order to test the value of network tree technology in this FIP development activity, conditions were selected in which the system was not responding as designed. For this case, the pitch axis lamp could not be reset after landing, and procedures were developed to determine why this condition existed.

3.3 FIP DEVELOPMENT (LRU Level). The next objective was to isolate the problem (pitch lamp fails to reset under all conditions) to the faulty LRU. In order to do so, a functional forest was developed for the pitch lamp and reset circuitry. This forest was also labeled with the appropriate connector, pin, and wiring data. This forest is illustrated in Figure 8. Using this forest, a set of FIPs was developed which used a combination of testing and observations to isolate the faulty LRU wiring. The voltage levels denoted in the procedures designed from Figure 11 do not reflect any tolerances.

The following steps represent one of the possible paths of this procedure (illustrated in Figures 9, 10, and 11):

Step 1 - The user observes the reaction of the pitch axis lamp as the servo/electronics reset switch is positioned to the ELEC position. In this case, the pitch lamp is not affected, and the user proceeds to step 2.

Step 2 - The user measures the voltage between aircraft ground and Flight Control Computer (FLCC) connector 2712P108, pin 111. In this case the voltage measures 28VDC, indicating that the main landing gear weight on wheels circuitry (relay B) is functioning correctly. The user proceeds to step 8.

Step 8 - The user measures the voltage between FLCC connector 2714P111, pin 69 and aircraft ground, while the servo/electronics reset switch is held in the ELEC position. In this case, the voltage is 28VDC, indicating that the reset signal is being transmitted to the FLCC successfully. The user then proceeds to Step 11.

Step 11 - The user now disconnects FLCC connector P107 and observes the pitch lamp. In this case the pitch lamp went off, indicating that the lamp has not been shorted to ground outside of the FLCC, and that the cause of the problem lies within the FLCC.

In this example, two observations and two tests (four steps) were required to isolate the FLCC. The maximum number of steps which might be performed in this procedure is six. This is the case where the pitch lamp is being shorted to ground in the wiring between the FLCC and the FLCP.

3.4 FIP Development (Internal to an LRU). The previous set of procedures allowed the user to isolate the FLCC as the faulty LRU with only four steps. Replacing that FLCC would make the aircraft operational again, with no additional equipment changeout other than the faulty LRU. The problem now is to determine, again as rapidly as possible, where the problem lies within the faulty LRU. To do this, a more detailed forest of the applicable circuitry within the FLCC was generated. This forest is illustrated in Figure 12.

This forest was generated to the card level. Each card is treated as a separate "box," and the interconnections between these cards are illustrated, along with the voltage levels or signal values at the appropriate points. The network trees used to develop this forest did not contain the connector or pin

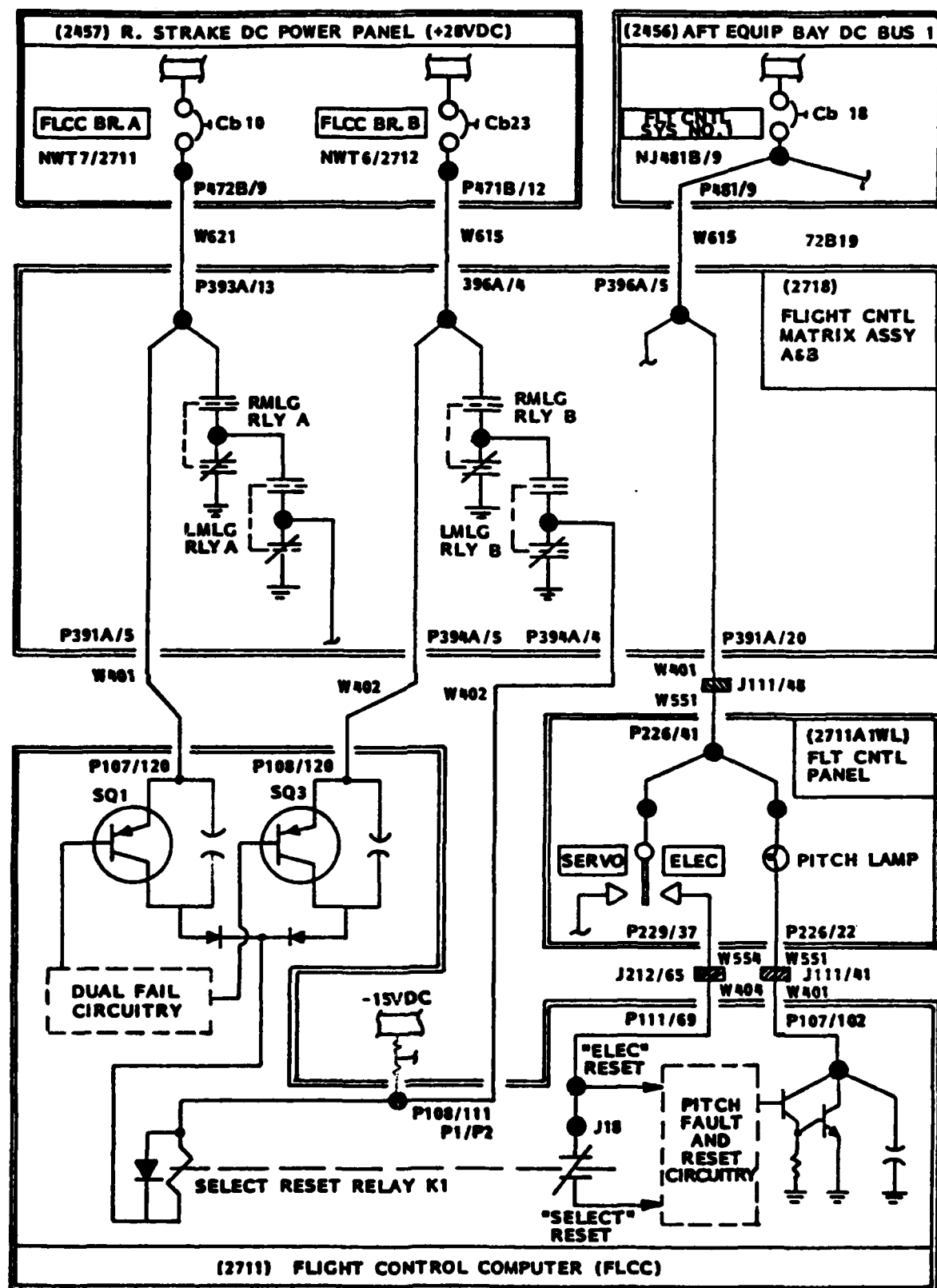


Figure 8. Functional Forest for Pitch Lamp and Reset

CONDITION: PITCH LAMP IS ILLUMINATED; CANNOT BE RESET ON THE GROUND.

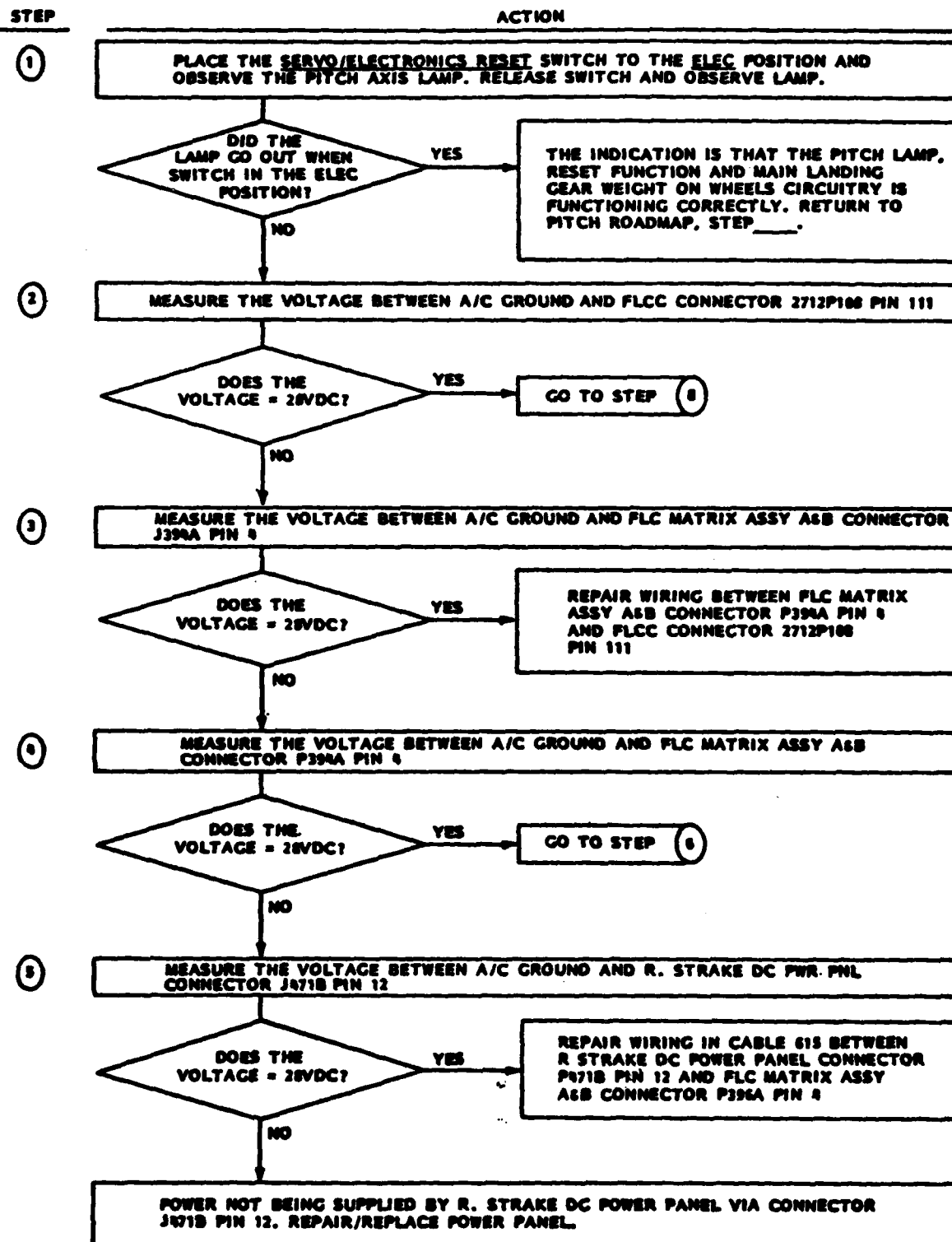


Figure 9. Pitch Axis Fault Isolation Procedure 2740-A1, Page 1

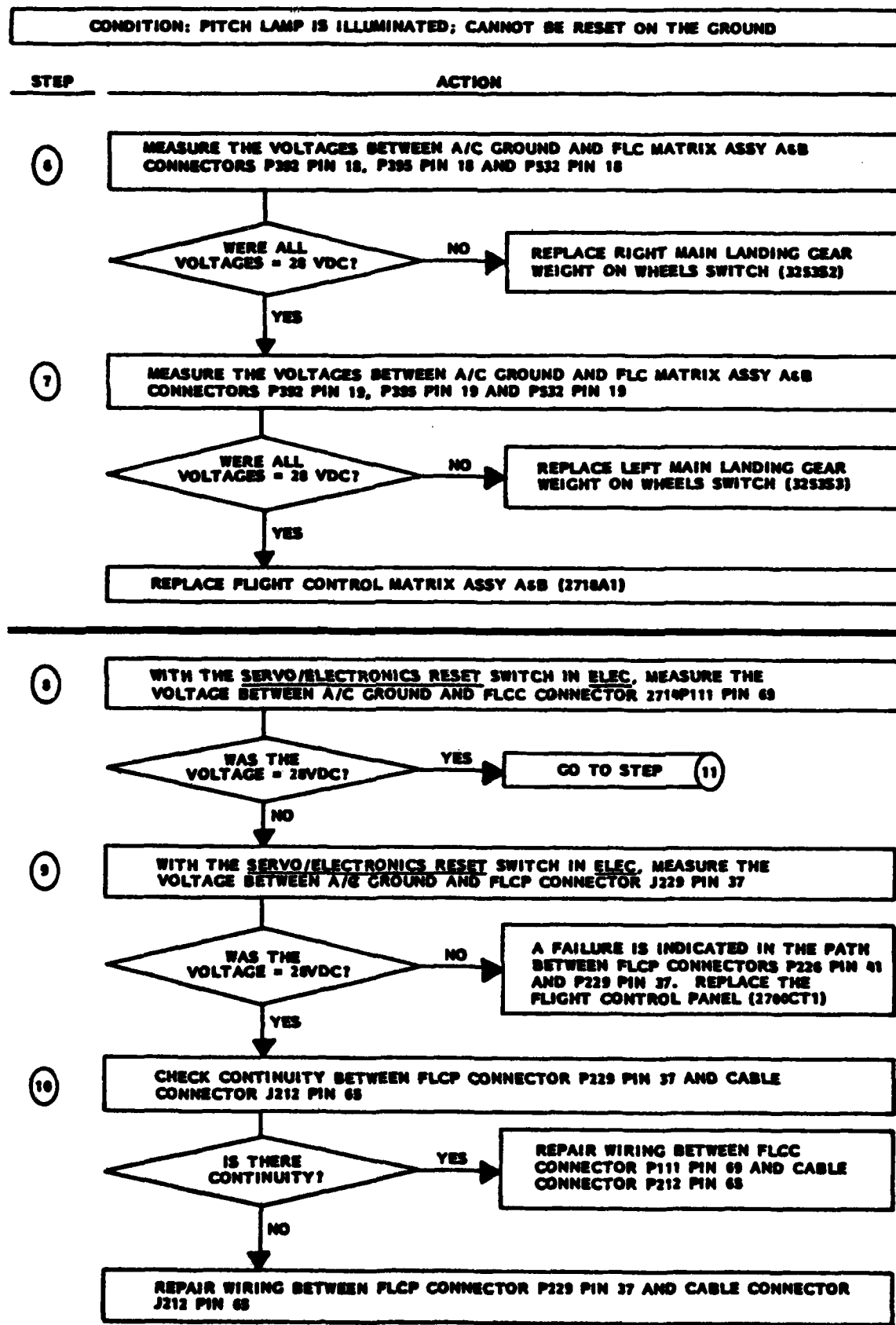


Figure 10. Pitch Axis Fault Isolation Procedure 2740-A1, Page 2

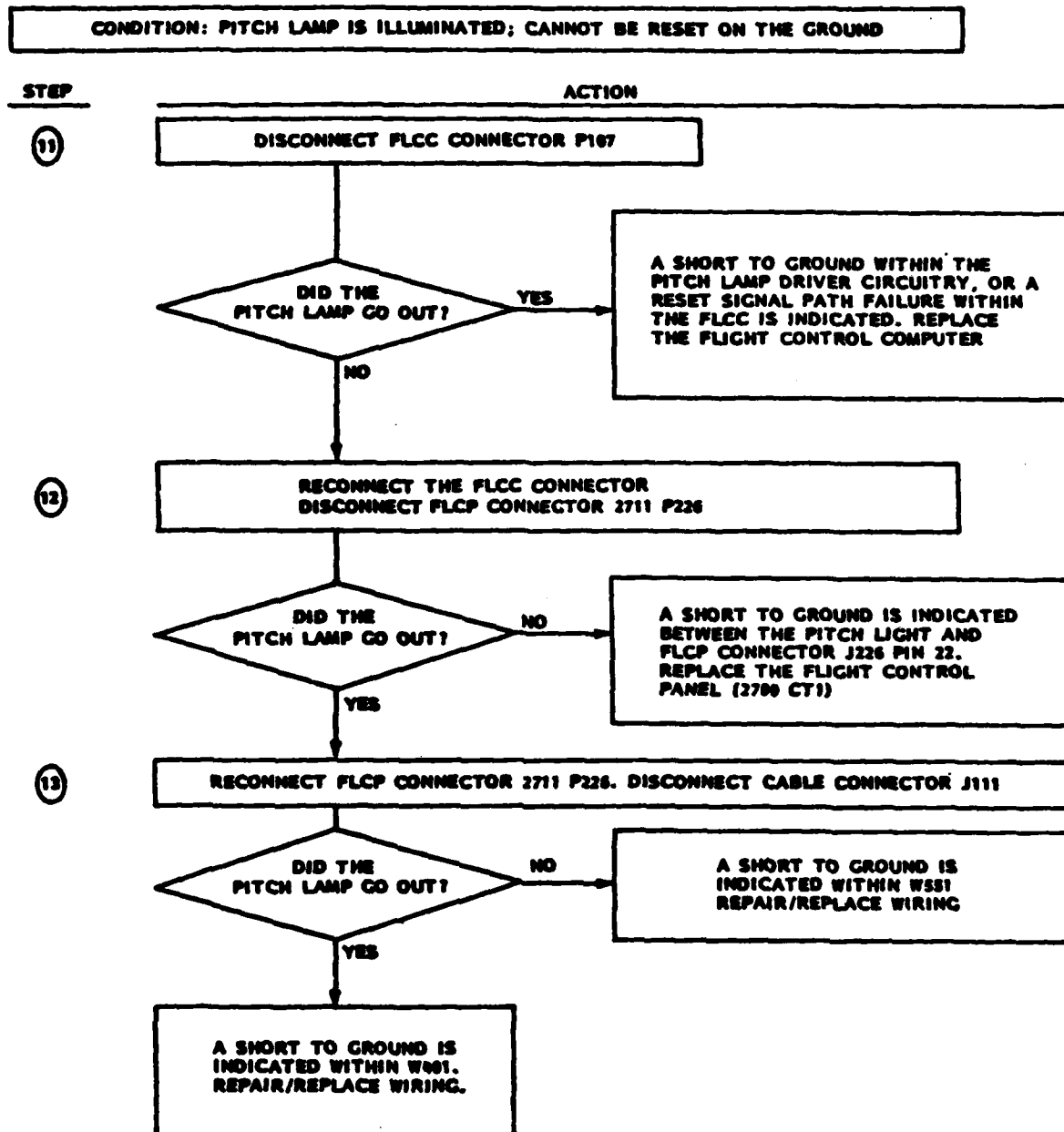


Figure 11. Pitch Axis Fault Isolation Procedure 2740-A1, Page 3

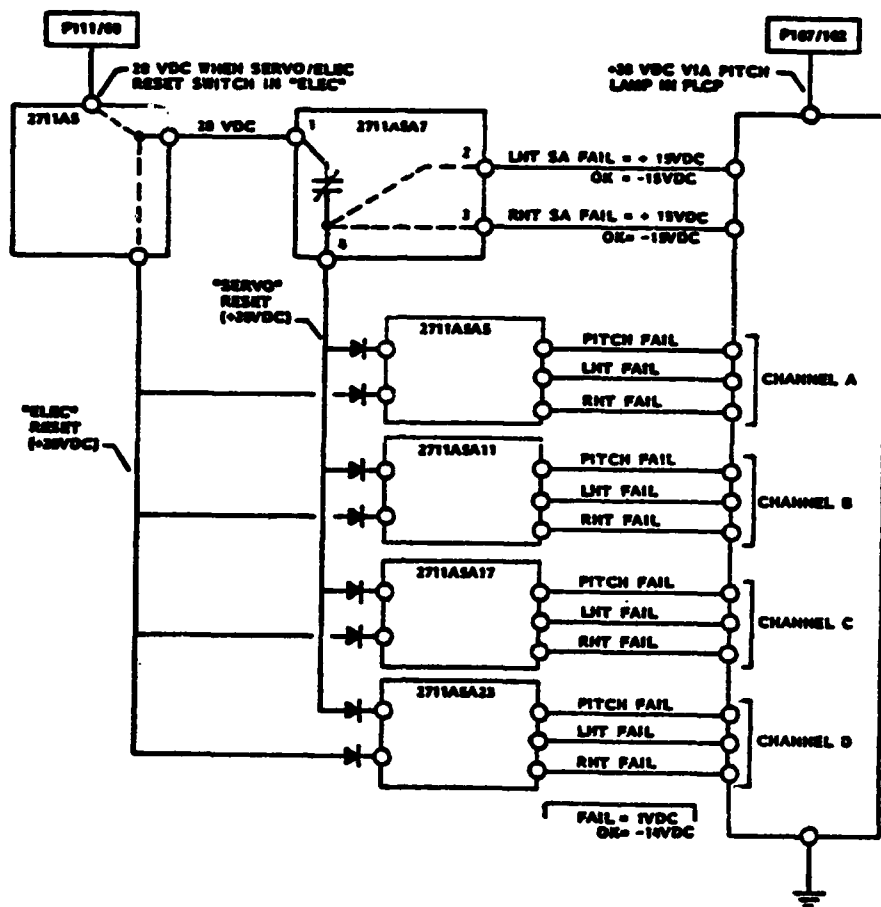


Figure 12. FLCC Internal Pitch Lamp Drive/Reset Forest.

information at this level. Network trees created for FIP development, of course, would have this information.

This forest can then be used to fault isolate to the card level. As an example, if 28VDC is applied to P111/69 and a measurement is taken between ground and pins 2, 3, and 4 (arbitrarily assigned) on card 2711A5A7, all three measurements should be 28VDC. If all three measure 0 volts, a measurement should be taken at pin 1 of the same card to determine if 28VDC is being applied there. If so, then the select reset relay or the path between pin 1 and the relay should be examined to determine the problem. This can be done by generating a forest of the appropriate circuitry on card 2711A5A7 and building procedures from that forest. The time available for this study did not allow for the development of forests and procedures beyond this level of detail.

4. FIP DEVELOPMENT METHODOLOGY

The following paragraphs define the methodology for generating FIPs using network tree technology. This methodology is based on the lessons learned during this study contract and includes those areas of automation which have been determined to be relatively simple to implement as well as highly cost effective.

4.1 Data Base Development

4.1.1 Data Requirements. The hardware data required for FIP development are the same as the data required to perform SCA and include integrated and detailed schematics, wire lists, wiring diagrams, and unique component specifications. In addition, system operating instructions, system descriptive data, and system specifications and design criteria are required (as available).

It is important to note that all system hardware data must be included. Subsystems cannot be selectively excluded from the analysis since all cause-and-effect relationships must be thoroughly investigated.

If the system being analyzed uses software in the control and/or monitoring of its activities, sufficient data pertaining to the purpose, function, and design of this software must also be supplied. Although it is not mandatory to perform a software analysis, the role of the software must be understood. This is especially true, in those cases where fault indications can be software driven.

When possible, systems operation data must be supplied that allow a complete understanding of the modes of operation and the system's configuration during these operational modes. This information is required in order to relate the system configuration to the possible fault indications as a function of the mode of operation.

4.1.2 Data Entry Process. The electrical configuration data are entered into the computer using an interactive data entry process which helps structure the data as 80 character pseudo card images as described in Section 5. Error checking software and a rigid quality control program are required to ensure data accuracy and completeness.

Each data record must contain at least the following information.

1. "From" Point: Location of the node (LRU, Card, etc.), the component type (resistor, switch, relay, etc.), and the appropriate pin number (as required).
2. "To" Point: Same as for the "From" Point.
3. Diode indicator to show current flow.

4. Remarks field to highlight special information such as signal type, power level, key element notations (reference Section 5.8), key element names, connector and pin terminology, and wire bundle nomenclature.

These data can be entered in any format that is compatible with the user's downstream software programs. The format described here reflects the structure required for compatibility with the Boeing family of SCA programs.

The data base must then be analyzed to determine whether there are any open ends and, if so, whether those open ends are valid. Most of this task should be automatic by use of audit and edit programs.

4.1.3 Change Tracking. Whenever system changes are implemented, these changes must be reflected in the data base. Changes to the system baseline are made interactively, and the network trees are also modified. Small changes may require only manual tree modifications. Large changes and some basic design changes may require that the appropriate network trees and forests be redrawn.

In certain cases, a system is designed so that it may exist in several basic configurations. In these cases, a baseline configuration data base can be developed, along with a separate "Delta" data base for each unique configuration. This approach also is useful when analyzing a system that interfaces with a variety of "Add On" components. Whether the data base is designed as a single data base or a baseline with "Deltas," it cannot be overemphasized that changes to the system must be tracked and reflected in the data base, and implementation of changes must be done throughout the system's life cycle to ensure that the resulting FIPs retain their accuracy.

4.2 Forest and Network Tree Development

4.2.1 Data Base Sorting. For each key element, a search is initiated to create a sub-data base which contains all data records that lead to and from the specific key element. Data records used in more than one sub-data base are keyed to indicate this multiple usage. Each sub-data base is then sorted by its reference designator (REFDES) in order to separate the data records into their physical locations (LRU, Card, etc.). As long as each key element has been flagged in the data base, and the REFDES of each data record has been properly assigned, then this entire sorting process can be performed automatically.

4.2.2 Forest and Tree Plotting. Once the sorting process has been completed, forest and network tree construction may begin. For each key element, reports are developed that define all of the LRU-to-LRU interfaces, the LRU-to-Card interfaces, the Card-to-Card interfaces, etc.

Assuming that the appropriate pin, connector, wire number, and signal value/power level information has been encoded in the data records, these reports will provide all of the system data required to construct the forests and

network trees used in the FIPs development process. Ideally, these forests and network trees should be created directly from these reports and be automatically plotted and annotated. Interactive user intervention should also be available to allow forest and network tree restructuring required for analysis and/or documentation purposes.

It is not mandatory that this plotting and annotation activity be fully automated. If this activity is performed manually, however, it is vital that the reports used to create these plots contain all the vital information (as described in Section 4.1.2).

4.3 Key Element Analysis

4.3.1 Fault Indicator Analysis. The forests and network trees for each fault indicator must be thoroughly analyzed to determine all the conditions which will drive the fault indicator. In addition, each fault indication must be classified as to whether it reflects a normal or an abnormal system response to a given set of conditions. All conditions, under which two or more fault indications are generated, must be defined. This fault/response information is vital to ensuring that maximum use of available fault clues is made. As the use of available fault clues is increased, the requirements for testing hardware are decreased. This is especially true of testing to the LRU level, although it can be shown to be true to even greater levels of detail.

4.3.2 Input Element Analysis. The forests and network trees for each key element that provides an input to the system (switches, sensors, etc.) or that changes the electrical configuration of the system must be thoroughly analyzed to determine all normal and abnormal system responses (including lack of response) to these inputs. Each abnormal response is also a valid fault indication and is added to the list of fault indications along with the conditions under which each abnormal response can occur.

4.3.3 Systems Operation Analysis. Additional fault indications may be determined through the analysis of the system operation. Network tree technology does not play an active role in this analysis, although it will play an important role later on, when the FIPs are developed for these fault indications.

4.4 Logic Diagram Development

4.4.1 Forest Modification. For each fault indication, the associated forests and network trees are analyzed to isolate the possible conditions which could stimulate that indication. In some cases, several forests (or portions of forests) must be combined to yield a more effective fault isolation tool. All data which do not relate to the fault indication being studied are eliminated from the modified forest. Since the initial goal is to isolate the fault to the LRU level, the modified forest does not include any more data than required for LRU isolation. When the goal is to isolate faults within all LRUs, the data included in the forest increase in detail, and extraneous data are eliminated.

Regardless of the level of detail of the forests, the goal is the same: To provide a clear picture of the circuitry involved with the fault indication, in a format which is readily usable in the development of testing logic diagrams.

4.4.2 Testing Logic Development. The modified forests are now in a condition that will allow easy determination of which pins and/or connectors should be tested, the correct voltage levels at those pins and/or connectors, and the conditions under which the correct voltage levels will and will not exist. Generating the testing logic is not a simple translation of the forest into a set of tests and observations, however. It accomplishes nothing to develop FIPs which minimize the number of tests performed, if the maintenance workload required to perform those tests is inordinately increased. Therefore, the testing logic diagrams must be developed not only through the use of the modified forests, but through the application of common sense, systems knowledge, and an awareness of the techniques and problems of the maintenance function.

The testing logic begins either at the LRU, where the majority of the tests can be performed, or at a location that will eliminate a large segment of the suspect circuitry.

Each subsequent test should then eliminate up to half of the remaining circuitry, until the fault is successfully isolated. In some cases, the suspect circuitry can be blocked into functional groups, and each test should be calculated either to eliminate or to fault isolate a specific circuitry group. This approach is highly beneficial when additional clues and visual observations might be available to help the fault isolation process. Consider the FIP illustrated in Figures 9, 10, and 11. If another axis lamp been illuminated (roll, for example) and that lamp successfully reset, then the user could safely proceed, beginning at step 11 of the procedure, and isolate the fault within three steps.

Once the fault has been isolated to the LRU, it is important that all the knowledge gained during this isolation process be passed on to the personnel that will perform the LRU-internal fault isolation. This can be accomplished readily by denoting the title of the appropriate LRU-internal FIP to be followed, directly on the LRU FIP at the step where the fault isolation is accomplished. In this way, shop maintenance can be initiated at the correct point in the fault isolation process, thereby greatly increasing the effectiveness of the total maintenance function.

By using the network trees and forests, FIPs can be developed to isolate a faulty LRU and then direct the user to the correct procedure set required to fault isolate the card and/or component, as applicable. Therefore, both line maintenance and shop maintenance procedures are developed together, in harmony, rather than as two separate, hard-to-relate entities.

5. CONCLUSIONS

5.1 Study Results

The technique of developing FIPs using network tree technology has the potential to be highly effective. The current data structure used for SCA is well suited for the searching, partitioning, and sorting required by this FIP development methodology. The automation of this searching, partitioning, and sorting does not appear to be a difficult task and would highly increase the productivity of the technique. The plotting of network trees and forests can be performed manually, although automated plotting (with interactive user intervention capability) is the preferable approach. Research and development activities are in progress to develop this capability.

Analysis of the resulting network trees and forests, and the translation of that information into effective FIPs, is still a manual process requiring skilled personnel. A great advantage to the use of forests and network trees, however, is the capability to create both line and shop maintenance procedures in which the continuity of the fault isolation process is maintained. This continuity between line and shop maintenance procedures should dramatically increase the effectiveness of the maintenance function.

Because the configuration of the system is contained in a computer data base, system changes can readily be reflected in that data base. The addition of peripheral devices to the system can also be easily accommodated. The system changes and additions can then be evaluated for impact on the FIPs (as well as system operations). The data base, therefore, becomes an important tool to be used in combatting maintenance problems that are generated by production phase and post-production phase system modifications.

5.2 Cautions

The success of this technique is highly dependent on the quality and content of the data base being used. This data base must contain every link in the system's electrical continuity. Each data record must describe a single point-to-point path. All system components must be included. Precise system nomenclature must be used. The reference designators must be accurately defined so that both the function and physical location of the circuitry can be identified.

The accuracy of the data base is dependent on the accuracy of the detailed schematics and other data required to build it. Changes must be incorporated as soon as possible. Drawing errors must be eliminated so that they do not influence the final data base.

The automated pathing, described in Section 4.2, cannot be performed without user intervention in some cases. Certain digital logic devices, and the use of software or firmware to perform systems functions, make end-to-end pathing currently impossible without an intervening analysis of the functions and

drivers of these devices. Present techniques will allow automated pathing up to, and away from, these devices. A trained analyst is required to provide the remaining links. It is envisioned that this problem can be partially alleviated by a "library" of component types and uses. The use of software network trees to provide the required hardware/software/hardware linkages is also a prime candidate for investigation.

It should be noted that the approach used in this study, and the resultant technique and areas for automation, were based on the use of a data base as defined by the Boeing SCA requirements. Similar type data bases can be used, providing they contain all of the data described in the previous paragraphs.

5.3 Technique Application

The technique defined by this study provides the tools required to develop FIPs in the form of network trees and forests. These tools can be used in the Logistics Support Analysis program to develop the actual fault isolation logic. Once developed, this logic would be formatted into document quality procedures by technical writing and documentation personnel.

APPENDIX A

DEFINITIONS OF KEY TERMS

The following are the key terms and definitions used throughout this Technical Report.

Data Record. The system configuration data required in the performance of Sneak Circuit Analysis (SCA) is entered into a computer in the form of data records. Each data record (as input) is 80 characters in length and has the generic structure shown in figure A-1.

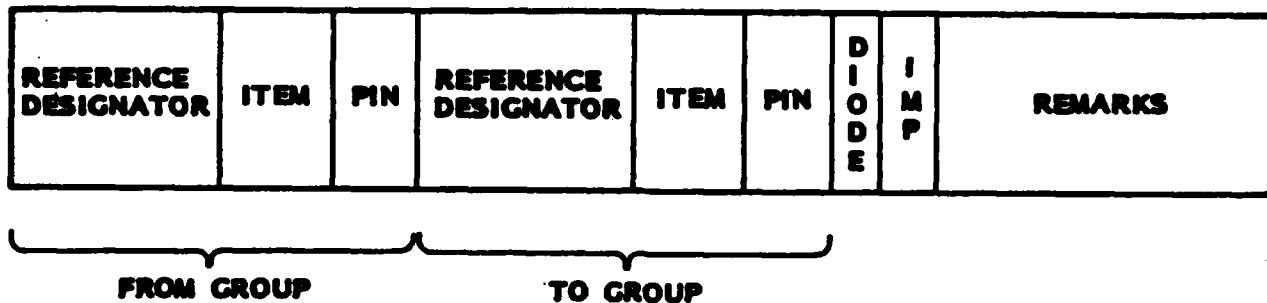


Figure A-1. Structure of an SCA Data Record

These data records are then searched and sorted by a variety of SCA applications programs in order to produce the network trees required to perform SCA.

Nodal Set. One output of the SCA family of programs is a report that contains groups of specific data records describing the circuitry which are functionally and physically linked. Each of these groups is called a nodal set and is given a unique number for future cross-reference.

Network Tree. The data in each nodal set report are then transferred to a topological plot of circuitry. Each of these annotated plots of the system circuitry is called a network tree. These network trees are then used directly in the performance of SCA. A typical network tree is illustrated in Figure A-2.

Forest. A forest is a plot of related network trees which provides a broader view of the physical and functional system continuities. Forests are constructed in various levels of detail in order to support both system operational analysis and the development of FIP logic diagrams. A high forest is illustrated in Figure A-3.

Power Forest. A power forest traces electrical power from the originating source to a major power distribution point. These forests are developed in order to analyze power distribution. Once this analysis is performed, these power forests are cross-referenced on the system signal forests and network trees by the number of the nodal set which displays the power distribution tree.

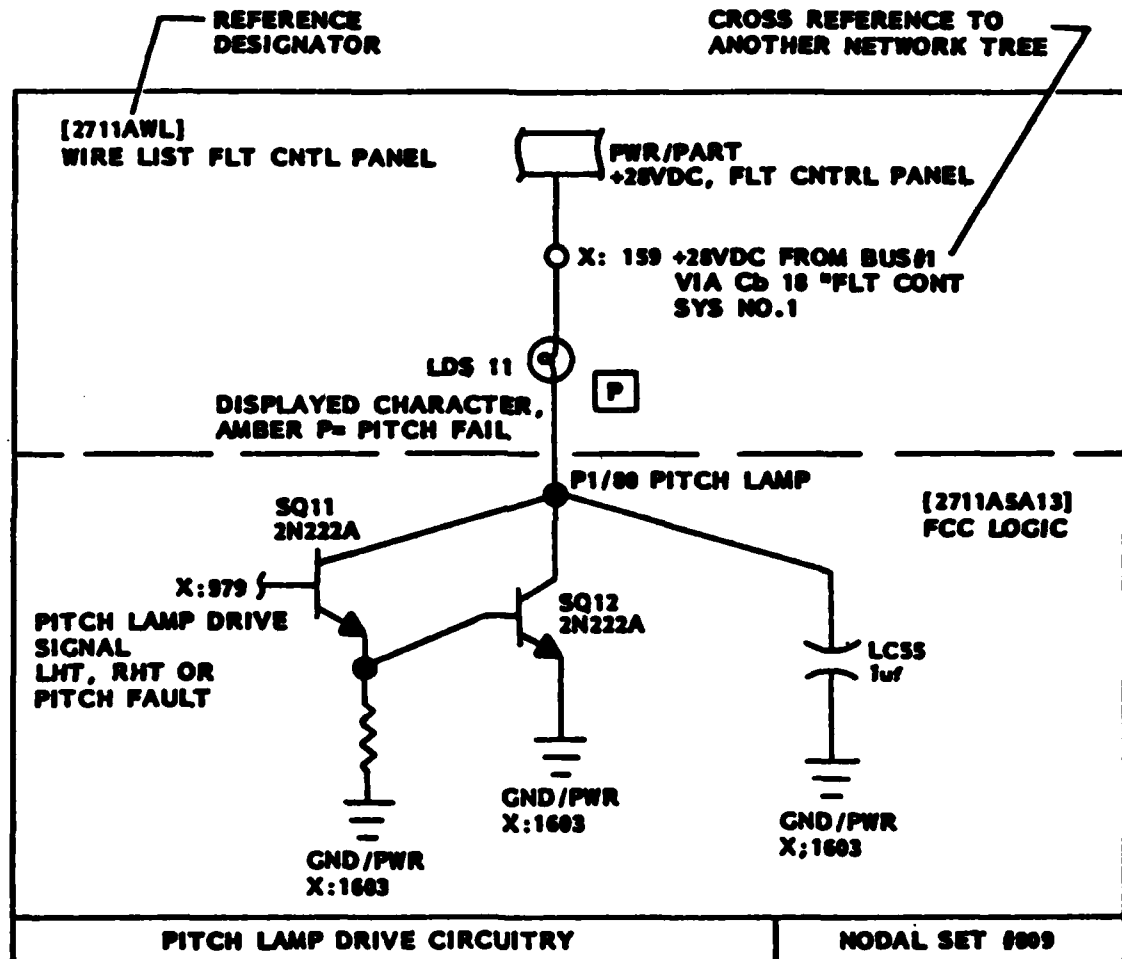


Figure A-2. Pitch Lamp Drive Circuitry Network Tree

Ground Forest. A ground forest traces electrical power from all points of connection to system ground. These forests are developed in order to analyze power returns. Once this analysis is performed, these forests are cross-referenced to the system signal forests as required.

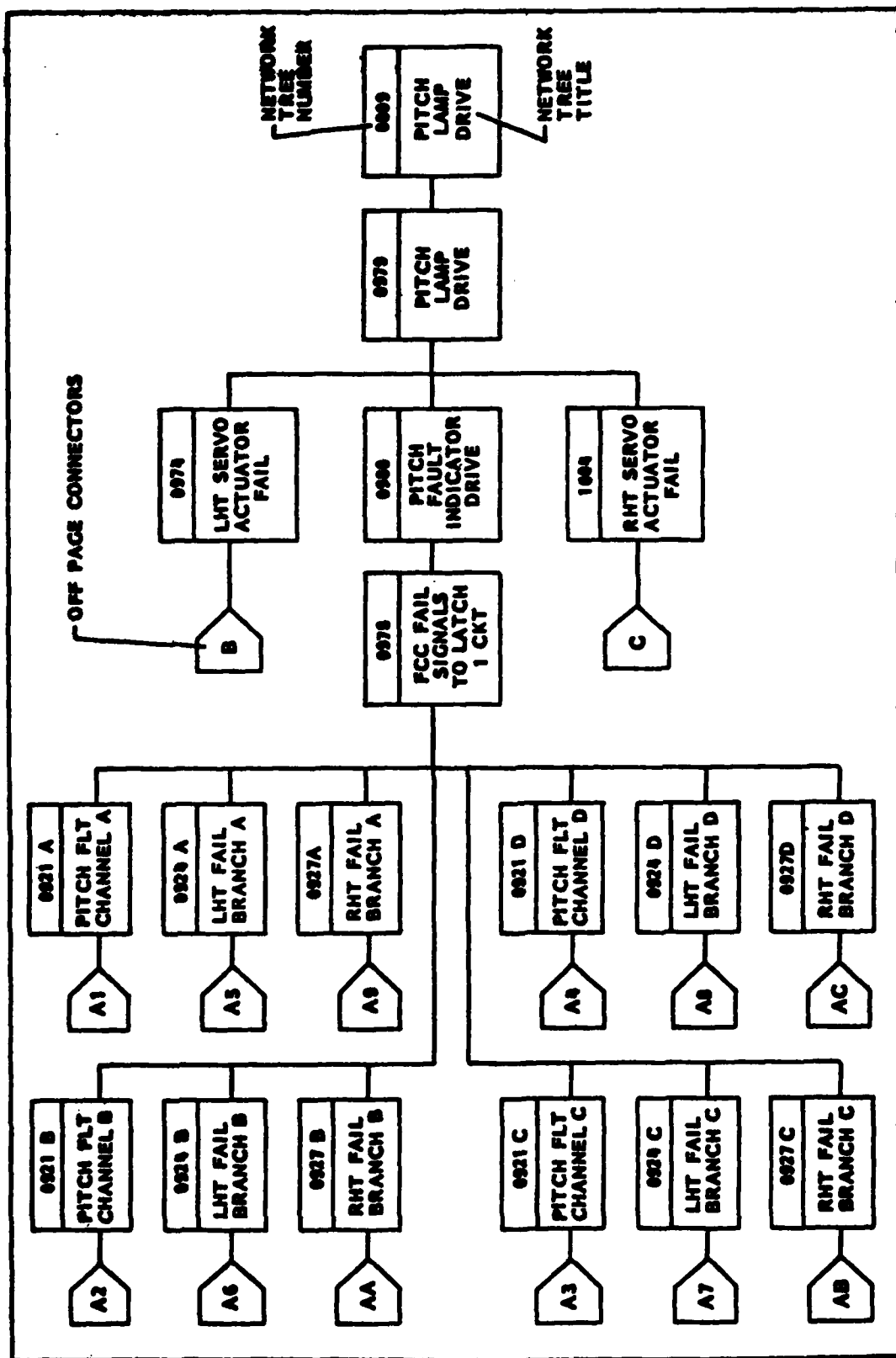


Figure A-3. Pitch Lamp Drive Overview Forest

Signal Forest. A signal forest traces electrical signal flow from point of origin to point of distribution. These forests represent the electrical system physical continuities. Functional continuities are attained through cross-referencing to other signal forests. In certain cases, both physical and functional continuities are represented in the same forest. The content of each signal forest and the level of detail are functions of the intended purpose of the forest. These forests can be combined and data selectively omitted when the purpose is FIP logic diagram development. No data are omitted when the purpose is system analysis.

Key Element. Each system component that interfaces with the environment is known as a key element. Examples of such key elements are lights, meters, flags, switches, controllers, and sensors. Each of these key elements either influences the operation of the system or is influenced by it. Those elements which output to the environment (lights, meters, flags, etc.) are normally the prime indicators of system malfunctions. Those elements which input from the environment to the system (switches, controllers, sensors) are also vital in that their failures to perform their intended functions are also key fault indications.

Reference Designator. Each SCA data record contains at least one reference designator (REFDES). The purpose of the REFDES is to provide the computer with exact component location information in a simple alphanumeric character string. For example, the first four characters could indicate the LRU, the next two a module within that LRU, and the the next two a printed circuit board, and so on.

Functional Pathing. Functional pathing is a technique in which the data base is searched, starting at a key element, and a collection of linked data records is obtained. It is the first step in isolating only those system continuities relating to the specified key element.

Logic Diagram. The logic diagram represents the sequential and conditional observation and testing steps required to isolate a specific malfunction. These diagrams are developed through the use of system forests and network trees and lead directly to the development of FIPs.